

Quorum Coding Challenge

Working with Legislative Data

Before You Start

Please make sure you read the entire document, including the last section which includes some suggestions, before getting started.

Overview

At Quorum, we collect and organize a large amount of publicly available government data. For example, we provide our clients the ability to visualize all of the bills that legislators voted for or against. To represent this data, we have the following important data models:

Person - An individual legislator elected to government. This includes everyone from President Joe Biden to Representative. David McKinley from West Virginia.

Bill - A piece of legislation introduced in the United States Congress.

Vote - A vote on a particular Bill. Bills can be voted on multiple times, as the Bill itself can undergo changes over the course of its life. For the purposes of this challenge, there will only be up to 1 Vote provided for each Bill.

VoteResult - A vote cast by an individual legislator for or against a piece of legislation. Each vote cast is associated with a specific Vote.

See the provided data for schema information for each of the data models.

Provided Data

You will be provided with a dataset comprised of the following four files:

bills.csv

Field	Type	Description
id	integer	The id of the bill
title	string	The name of the bill
Primary Sponsor	integer	The id of the primary sponsor (of type Person)

legislators.csv

Field	Type	Description
id	integer	The id of the legislator
name	string	The name of the legislator

votes.csv

Field	Type	Description
id	integer	The id of the Vote
bill_id	integer	The id of the bill that this vote is associated with

vote_results.csv

Field	Type	Description
id	integer	The id of the VoteResult
legislator_id	integer	The id of the legislator casting a vote
vote_id	integer	The id of the Vote associated with this cast
vote_type	integer	The type of vote cast - 1 for yea and 2 for nay

Deliverables

You will be provided with a list of legislators, bills, votes, and vote results as specified above. You'll be asked to answer the following questions:

1. For every legislator in the dataset, how many bills did the legislator support (voted for the bill)? How many bills did the legislator oppose?
2. For every bill in the dataset, how many legislators supported the bill? How many legislators opposed the bill? Who was the primary sponsor of the bill?

Your program should take in the data provided and output a CSV for each of the questions in Part 1. For example, you might name the first file **legislators-support-oppose-count.csv**. Each row in this CSV should represent a legislator and have the following columns:

Field	Type	Description
id	integer	The id of the legislator
name	string	The name of the legislator
num_supported_bills	integer	The number of bills the legislator voted Yea on from the dataset
num_opposed_bills	integer	The number of bills the legislator voted Nay on from the dataset

You might name the second file **bills.csv**. Each row in this CSV should represent a bill and have the following columns:

Field	Type	Description
id	integer	The id of the bill
title	string	The title of the bill
supporter_count	integer	The number of legislators that supported this bill in the vote for it
opposer_count	integer	The number of legislators that opposed this bill in the vote for it.
primary_sponsor	string	The name of the primary sponsor of the bill. If the name of the sponsor is not available in the dataset, the cell should be "Unknown"

After completing your implementation, you should include a write up that answers the following questions:

1. Discuss your solution's time complexity. What tradeoffs did you make?
2. How would you change your solution to account for future columns that might be requested, such as "Bill Voted On Date" or "Co-Sponsors"?
3. How would you change your solution if instead of receiving CSVs of data, you were given a list of legislators or bills that you should generate a CSV for?
4. How long did you spend working on the assignment?

You should send us a folder that contains the following:

- Source code for your implementation, including steps to run your code
- Output files, as requested
- A writeup that responds to the questions asked above

Evaluation Criteria

Your solution will be evaluated based on the following criteria:

1. **Correctness** - Is your output correct based on the provided data? How did you prove correctness?
2. **Structure/Readability** - Is your code well organized and easy to read? Can it be extended reasonably if new requirements are given?
3. **Proficiency with Language** - Does your code make use of typical conventions in your language of choice?

Suggestions

- You shouldn't need to spend more than 2-3 hours on this, so we recommend choosing a simple approach and ensuring it is well-formatted and correct.
- You are free to make use of the internet to look up syntax.
- You may use any programming language you like, though a high level language such as Python will probably be best suited. Feel free to make use of any external libraries (such as parsing CSVs), but you should not need anything complicated such as a database.

- Consider committing your work with git so that we can see your progress